Adaptive hybrid Metropolis-Hastings samplers for DSGE models

Ingvar Strid*Paolo Giordani† Robert Kohn‡

SSE/EFI Working Paper Series in Economics and Finance No. 724

February 2010

Abstract

Bayesian inference for DSGE models is typically carried out by single block random walk Metropolis, involving very high computing costs. This paper combines two features, adaptive independent Metropolis-Hastings and parallelisation, to achieve large computational gains in DSGE model estimation. The history of the draws is used to continuously improve a t-copula proposal distribution, and an adaptive random walk step is inserted at predetermined intervals to escape difficult points. In linear estimation applications to a medium scale (23 parameters) and a large scale (51 parameters) DSGE model, the computing time per independent draw is reduced by 85% and $65-75\%$ respectively. In a stylised nonlinear estimation example (13 parameters) the reduction is 80%. The sampler is also better suited to parallelisation than random walk Metropolis or blocking strategies, so that the effective computational gains, i.e. the reduction in wall-clock time per independent equivalent draw, can potentially be much larger.

Keywords: Markov Chain Monte Carlo (MCMC), Adaptive Metropolis-Hastings, Parallel algorithm, DSGE model, Copula.

*Department of Economic Statistics, Stockholm School of Economics. Email: ingvar.strid@hhs.se. Tel: +4687369232. Adress: Stockholm School of Economics, P.O. Box 6501, SE-113 83 Stockholm, Sweden.

†Research Division, Sveriges Riksbank, Email: paolo.giordani@riksbank.se. Adress: SE 103 37 Stockholm, Sweden. Phone: +4687870498.

‡Australian School of Business, University of New South Wales. Email: R.Kohn@unsw.edu.au. Phone: +61293852150. Adress: UNSW Sydney NSW 2052, Australia.

1 Introduction

Dynamic Stochastic General Equilibrium (DSGE) models are commonly estimated using Bayesian methods. A prior distribution for the model parameters is updated to a posterior distribution using likelihood information, with sampling from the posterior carried out using Markov Chain Monte Carlo (MCMC) inference. The main features of the approach are well described in the review article by An and Schorfheide (2007*a*) and some notable contributions to the field are Smets and Wouters (2003), Adolfson, Laséen, Lindé and Villani (2007) and Fernández-Villaverde and Rubio-Ramírez (2007). A key feature in this growing literature is the almost exclusive use of the single-block random walk Metropolis (RWM) algorithm to sample from the posterior distribution of the model parameters. As noted by An and Schorfheide (2007*b*) '...there is little research that tries to compare the performance of alternative sampling schemes and develop potentially better MCMC schemes' for DSGE models and Adolfson, Lindé and Villani (2007*a*) states that 'there is clearly room for big improvements in simulation efficiency'. Simulation efficiency is important because MCMC inference is very time consuming for DSGE models, which considerably slows down the process of model development.

The purpose of our paper is to evaluate adaptive MCMC algorithms applied to the estimation of DSGE models. The main element of adaptive sampling schemes is the use of previous MCMC draws for the design of efficient proposal densities. The samplers evaluated here are based on four main ideas. The first is to use the history of posterior draws to repeatedly estimate t-copula densities with mixture of normal marginals, and use these as proposal distributions in an independence Metropolis-Hastings sampler. The efficient performance of proposals constructed in this way has been established for a variety of models in Kohn and Giordani (2008) and Silva, Kohn, Giordani and Mun (2008). Second, in order to alleviate some potential shortcomings of a pure independence chain approach in high-dimensional problems, we propose simple hybrid, deterministic cycling, algorithms which occasionally use random walk proposals to escape points in the posterior parameter space where the posterior-to-proposal ratio is large. Third, the time per posterior draw does not increase significantly in comparison with the RWM sampler due to the fast estimation of mixture of normal and t-copula densities. Fourth, our preferred algorithms are very suitable for parallel implementation. Parallel computation is becoming increasingly accessible and has the potential to drastically reduce computing time in a variety of problems, and different MCMC schemes differ greatly in their suitability for parallel implementation.

The sampling performance of the competing algorithms is first evaluated using two linearised DSGE models: A modified version of the medium-scale DSGE model developed by Smets and Wouters (2003), one of the core models at the European Central Bank, and the large-scale open economy model, nicknamed RAMSES, in use at the central bank of Sweden (Adolfson, Laséen, Lindé and Villani (2007)). Based on a comparison of inefficiency factors the efficiency of the main sampler presented here is respectively

$6-7$ and $3-4$ times larger than for the standard single-block RWM algorithm in these estimation examples. Next the samplers are evaluated using a nonlinearly approximated small-scale New Keynesian model, where a particle filter is employed for the likelihood evaluation. In this case the efficiency of the main sampler is roughly 5 times larger than for the RWM algorithm. Taking parallelisability into account the total efficiency gain, as measured by independent equivalent draws per time unit, is potentially much larger in all these examples.

Recently Chib and Ramamurthy (2009) have suggested a tailored randomised-block (TaRB-MH) algorithm for the estimation of DSGE models. They show that the TaRB-MH algorithm is able to obtain the correct posterior distribution in two well-known DSGE model estimation problems where the single-block RWM algorithm fails to explore all local modes. The reported computational times, however, are a factor $150-1100$ times larger per posterior draw in comparison with the RWM algorithm, and the TaRB-MH algorithm is not suitable for parallelisation.

The sampling algorithms presented here are entirely different in character: they have roughly the same computing time per posterior draw as the single-block RWM and are easily parallelised. The emphasis is on reduced computing time per independent equivalent draw rather than on an increased ability to explore multiple modes. We choose to consider only single-block samplers in this paper because multiple block MCMC algorithms cannot be parallelised efficiently. Obviously, as the dimension of the parameter vector increases one-block independence samplers will accept less and less often, and at some point it will be necessary to apply blocking methods. However, the successful application to RAMSES, with its 51 parameters and 60 state variables, suggests that our single block algorithms should prove useful in a large number of problems.

The paper proceeds as follows. In section 2 the econometrics of DSGE models is briefly described. In section 3 the sampling algorithms are presented and in section 4 parallelisation of the algorithms is described on a conceptual level. In section 5 the statistical performance of the samplers is evaluated in three DSGE model estimation examples.

2 DSGE models

The equilibrium of a DSGE model is described by a set of nonlinear expectational equations. These consist of constraints and first-order conditions derived from the optimising behaviour of various agents, such as consumers and firms, in a model economy. The solution to the equations is given by a policy function

$$X_t = g\left(X_{t-1}, \epsilon_t; \theta\right), \tag{1}$$

which relates a vector of state variables X_t to its lagged value, a vectors of innovations ϵ_t, and the parameter vector θ. The policy function cannot typically be derived in closed

form. This implies that numerical approximation methods must be used to obtain an approximation $\hat{g}$ to g.

The likelihood evaluation for a DSGE model consists of two parts. First, given a parameter vector θ, an approximation to the policy function around the deterministic steady state is obtained (the solution) and represented as a state-space model. DSGE models with non-stationary variables, e.g. induced by a non-stationary technology shock, are transformed to stationarity prior to solution of the model, such that X_t is a vector of stationary variables. Second, filtering methods are used to evaluate the likelihood function.

We first consider (log-)linear approximations, the associated linear and Gaussian state-space (LGSS) model and Kalman filtering. Several methods are available to solve DSGE models linearly, e.g. Anderson and Moore (1985), Sims (2000) and Klein (2000). In general the fastest, and also most accurate, method available is the Anderson-Moore algorithm (AIM), see the comparison in Anderson (2008).

In the (log-)linear approximation case the resulting state-space model is

$$X_t = \hat{g}\left(X_{t-1}, \epsilon_t; \theta\right) = T\left(\theta\right) X_{t-1} + R\left(\theta\right) \epsilon_t \tag{2}$$

and

$$Y_t = d\left(\theta\right) + ZX_t + v_t,\ t = 1, ..., T\ , \tag{3}$$

where [2] is the state equation and [3] is the observation equation. Here X_t (dimension n_x) is a vector containing the state variables, in the econometric sense, and Y_t (dimension n_y) is a vector containing the observed variables. The parameters of the model are collected in the vector θ (dimension n_θ) and the coefficient matrices, T (which is typically dense) and R, and the vector d are nonlinear functions of θ. For the DSGE models considered in this paper Z is a selector matrix which does not depend on θ. The innovations, ϵ_t (n_ϵ), and the measurement errors, v_t (n_v), are assumed to be independent and normally distributed, $\epsilon_t \backsim N\left(0, \Sigma_\epsilon\right)$ and $v_t \backsim N\left(0, \Sigma_v\right)$. The resulting model is therefore a particular type of LGSS model. Importantly, in DSGE models the functions $T\left(\theta\right), R\left(\theta\right), d\left(\theta\right)$ are not available analytically (except in very special cases), but are obtained for a given θ by numerically solving for the rational expectation equilibrium, so Gibbs sampling is not feasible in general. The distribution of the initial state vector, X_0, is typically assumed to be equal to the stationary distribution of the state vector, $X_0 \backsim N\left(0, \Sigma_x\right)$, where Σ_x is the solution to the Lyapunov equation

$$\Sigma_x = T\Sigma_x T^T + R\Sigma_\epsilon R^T. \tag{4}$$

In the nonlinear estimation example of this article a second order approximation to the policy function, [1], is used, following Schmitt-Grohe and Uribe (2004) and Klein (2000). The approximate solution can be cast in the following state-space form. The state equation vector is separated into an equation vector for the exogenous state variables (the shocks)

$$X_{1t} = AX_{1t-1} + \varepsilon_t, \tag{5}$$

and an equation vector for the endogenous predetermined variables and a subset of the nonpredetermined variables of the model

$$X_{2t} = B\tilde{X}_{t-1} + Cvech(\tilde{X}_{t-1}\tilde{X}_{t-1}^T) + e, \tag{6}$$

where $\tilde{X}_{t-1} = (X_{1t}^T \; X_{2t-1}^T)^T$ and $X_t = (X_{1t}^T \; X_{2t}^T)^T$. If we let $C = 0$ and $e = 0$, [5] and [6] reduce to [2], and partitioning $B = [B_1 \; B_2]$ the state transition matrix in [2] is

$$T = \begin{pmatrix} A & 0 \\ B_1 A & B_2 \end{pmatrix}. \tag{7}$$

Having expressed the nonlinear state equation in this way, the measurement equation is linear and given by [3]. In the nonlinear case the normality assumption for ϵ_t and v_t is no longer crucial but we retain it here for convenience.

The likelihood function is

$$p(Y_{1:T}|\theta) = \prod_{t=1}^{T} p(Y_t|Y_{1:t-1};\theta) = \prod_{t=1}^{T} \int p(Y_t|X_t)p(X_t|Y_{1:t-1})dX_t,$$

where $Y_{1:T} = (Y_1, ..., Y_T)$ is the data. In the (log-)linear case the likelihood is evaluated using the prediction error decomposition and the Kalman filter, e.g. Harvey (1989). In estimating large DSGE models, and assuming that the fastest solution method (AIM) is used, the dominant share of computational time is spent on Kalman filtering. DSGE-specific Kalman filter implementations, mainly based on exploitation of the special structure of the state transition matrix T in [7], can reduce computing time significantly in the case of large models (Strid and Walentin (2008)).

In the nonlinear case a standard particle filter (SPF) is applied for the likelihood evaluation, see Arulampalan, Maskell, Gordon and Clapp (2002) for an introduction to particle filtering and Fernández-Villaverde and Rubio-Ramírez (2007) and An (2005) for applications to DSGE models. The SPF requires two conditions to be fulfilled: the ability to simulate from the state equation, [5] and [6] , and the ability to evaluate the observation density, $p(Y_t|X_t)$. The filter yields an unbiased estimate, $\hat{p}(Y_{1:T}|\theta)$, to the likelihood, $p(Y_{1:T}|\theta)$. The computational burden associated with the particle filter implies that the demand for efficient sampling schemes is even larger in this case. For the small-scale New Keynesian model used for the illustration in section 5.3 the particle filter likelihood evaluation, with $N = 40,000$ particles, is more than 1000 times slower than the Kalman filter evaluation for the corresponding linearised model.

The kernel of the posterior density of the parameter vector θ is

$$p(\theta|Y_{1:T}) \propto p(Y_{1:T}|\theta)\tilde{\pi}(\theta)I(\theta \in \Theta_D),$$

where attention is usually restricted to the determinacy region, Θ_D (see Lubik and Schorfheide (2004) for an exception). In the nonlinear case the exact likelihood is replaced by the particle filter approximation. The determinacy region is the subset of the

parameter space where the model has a unique and stable solution. (DSGE models typically admit an infinite number of explosive solutions.) Here we interpret the restriction as being part of the formulation of the prior distribution, i.e. the prior density π is truncated at the boundary of the indeterminacy region. The effective prior distribution is typically formulated as a set of marginal prior distributions[1]*:*

$$\pi(\theta) = I(\theta \in \Theta_D)\tilde{\pi}(\theta) = I(\theta \in \Theta_D)\Pi_{j=1}^{n_\theta}\pi_j(\theta^j),$$

where θ^j is the j^{th} element of the vector θ.

There are essentially three categories of structural parameters: unbounded parameters, parameters which are bounded from below or above, and parameters which are bounded both from below and above. In the DSGE literature, parameters belonging to these classes are typically given normal, gamma (or inverse gamma) and beta prior distributions, respectively.[2] Parameters with a (inverse) gamma prior density on $[0, \infty)$ are commonly reparameterised using the log transformation, and for parameters with beta prior densities on $[0, 1)$ the logit transformation is used, see e.g. Adolfson, Laséen, Lindé and Villani (2007).

Reparameterisation serves two purposes. First, the transformation to an unbounded parameter space simplifies optimisation of the posterior, which is often used as a preliminary step for MCMC analysis. Second, the posterior of the transformed set of parameters is typically closer to a normal distribution in shape, and therefore simpler to sample from. Reparameterisation is important in order to avoid exaggerating the gains of more elaborate MCMC approaches, such as those presented below (in section 3), and furthermore it is easy to implement (Adolfson, Lindé and Villani (2007*a*)).

Having obtained the posterior density of the parameters $p(\theta|Y_{1:T})$, the posterior density of the state variables $p(X_{1:T}|Y_{1:T})$ can be obtained via smoothing techniques using the relation

$$p(X_{1:T}|Y_{1:T}) = \int p(X_{1:T}|\theta, Y_{1:T})p(\theta|Y_{1:T})d\theta,$$

where $X_{1:T} = (X_1, ..., X_T)$, see e.g. Durbin and Koopman (2001) in the linear case. It is useful to contrast the 'marginalisation' approach used in the case of DSGE models, where $X_{1:T}$ is integrated out using the Kalman filter, with the classic two-block scheme for sampling in more 'typical' LGSS models, where one alternately samples from densities $p(\theta|X_{1:T}, Y_{1:T})$ and $p(X_{1:T}|\theta, Y_{1:T})$ using Gibbs and/or Metropolis-Hastings updates. The inability to sample from $p(\theta|X_{1:T}, Y_{1:T})$ distinguishes the DSGE-LGSS model from other LGSS models.

[1]The marginal priors are not, strictly speaking, independent due to the indeterminacy constraint.

[2]In all models considered in this paper the parameter describing the steady state technology growth rate has a truncated normal prior distribution. However, the mass attached to the truncation region is small enough to be disregarded, e.g. in the case of the RAMSES model the distance between the prior mean and the truncation point is 12 standard deviations.

3 Sampling algorithms

The objective of sampling algorithms is to generate a sequence of draws, θ_i, $i = 1, ...R$ from the posterior kernel $p(\theta) = p(\theta|Y_{1:T})$, where R is the length of the chain. In the context of Bayesian estimation of DSGE models the single-block random walk Metropolis (RWM) algorithm has been the preferred sampling method. In the RWM algorithm a proposal θ_p is generated using a symmetric proposal density $q(.|\theta_i)$, where θ_i is the current state, and accepted with probability

$$\alpha_{i+1} = \min\left\{1, \frac{p(\theta_p)}{p(\theta_i)}\right\}.$$

The proposal distribution is typically a normal distribution $q(\theta_p|\theta_i) = N(\theta_i, \Sigma)$, with Σ proportional to the inverse of the Hessian at the posterior mode, θ_m, $\Sigma = -\varkappa H_m^{-1}$ where $\varkappa > 0$. A crucial step in this approach is then the optimisation of the posterior density since the quality of the RWM sampler will rely on the quality of the estimated inverse Hessian at the posterior mode. As demonstrated starkly by both Andreasen (2008) and Chib and Ramamurthy (2009) finding the global posterior (or likelihood) mode is a non-trivial problem in large-dimensional DSGE models. An adaptive RWM approach (see below) is potentially more robust; because the matrix Σ is updated throughout the chain, a poor starting value for Σ does not affect sampling efficiency as heavily.

There are several reasons for the popularity of the single-block RWM algorithm in the DSGE model context. First, a standard $B-$block Metropolis-Hastings sampler is automatically penalised by a B-factor increase in computational time. Furthermore it is non-trivial, at least a priori, to group parameters into fixed blocks such that there is weak dependence between parameters in separate blocks. Second, for large dimensional parameter vectors the independence Metropolis-Hastings (IMH) is fragile in the sense that it can occasionally get trapped for long spells at points in the parameter space where the posterior-to-proposal ratio, $p(\theta_i)/q(\theta_i)$, is high. For simplicity we will refer to such occurrences as 'difficult points'. The acceptance rate of the IMH sampler is

$$\alpha_{i+1} = \min\left\{1, \frac{p(\theta_p)/q(\theta_p)}{p(\theta_i)/q(\theta_i)}\right\},$$

where $q(\theta)$ denotes the independent proposal density.

The samplers used in this paper arise from a general deterministic hybrid formulation. Proposals are generated from the densities $q_{j,i+1}(\theta_p|\Theta_i)$, $j = 1, 2, 3$, according to a deterministic cycling scheme where $\Theta_i = \{\theta_0, ..., \theta_i\}$ denotes the draws up to iteration i and θ_0 is the starting value of the chain. The fraction of draws from the proposal distributions are denoted α_j, $j = 1, 2, 3$, and these fractions may change at prespecified points of the chain.

The first component, q_1, is a random walk Metropolis (RWM) proposal and the second component, q_2, is a unimodal and symmetric independence proposal (UIMH), in our

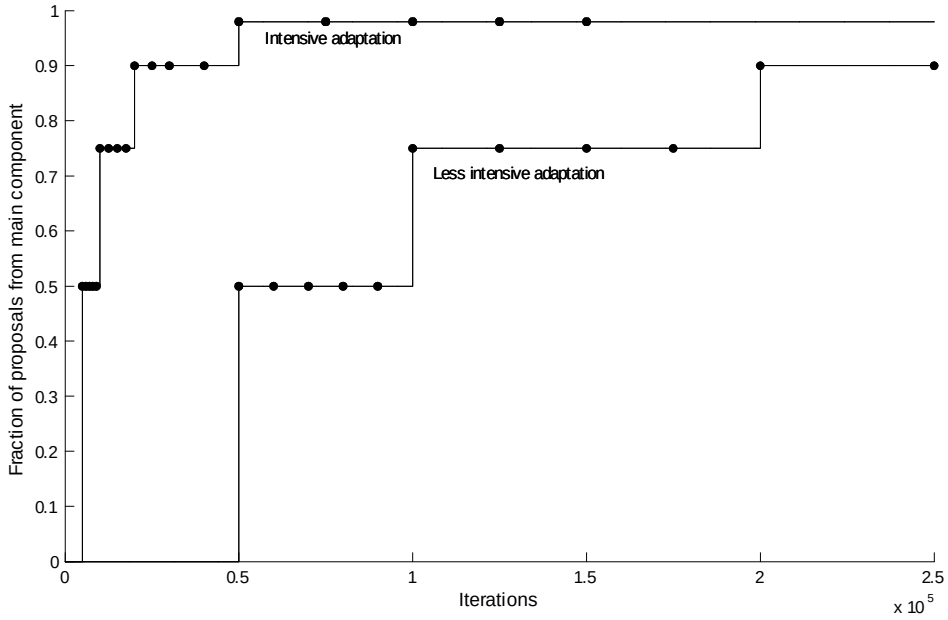
experiments a multivariate t or normal distribution. These proposal distributions may be adaptive or non-adaptive. The *main component*, q_3, is a t-copula with mixture of normal marginal distributions estimated using clustering methods (TC) as in Silva, Kohn, Giordani and Mun (2008). The t copula provides a fast and flexible way of estimating a multivariate density and in general it yields a more accurate estimate of the posterior density than the 'standard' symmetric independence proposal, q_2. The components are described in detail below. Various samplers thus arise by different choices of α_j, e.g. the RWM sampler is described by $\alpha_1 = 1$ and $\alpha_2 = \alpha_3 = 0$.

The *main sampler* of interest, denoted RWM-UIMH-TC, is a sampler containing all three components where the fraction of draws from the main component q_3 increases gradually through the chain. In an initial stage $\alpha_3 = 0$ and then α_3 is increased to some value $\alpha_3 = \alpha_3^* \geq 0.75$, such that eventually the majority of draws are generated by the main component. For simplicity the ratio of random walk proposals to unimodal independence chain proposals (α_1/α_2), is kept fixed throughout the chain.

In the main sampler one function of the first two components is to start up the chain and produce a number of accepted draws in an initial stage of sampling. A pure independence sampler, i.e. a sampler using some combination of q_2 and q_3 only, runs the risk of getting trapped for long spells at difficult points in the parameter space, especially as the dimension of the problem increases. The primary motive to include the random walk component also *after* an initial phase is therefore to be able to escape such difficult points. However, the improved approximating ability of q_3 in comparison with q_2 should allow us to insert these RWM steps less frequently.

The *intensity of adaptation* depends on (i) how early adaptation begins, (ii) the frequency of adaptation, i.e. how often q_3 is re-estimated, and (iii) how quickly the share of draws from the main component (α_3) increases. The *adaptation schedule*, the set of points where q_3 is re-estimated, is denoted $A = \left\{i_0^A, i_1^A, i_2^A, ..., i_M^A\right\}$ where $i_0^A = 0$. In order not to complicate the sampler further these are also the points where we allow the shares α_j to change, such that there is a *hybrid schedule* $H = \left\{\alpha_3^0, \alpha_3^1, \alpha_3^2, ..., \alpha_3^M\right\}$ corresponding to the adaptation schedule. Here α_3^j denote the fraction of draws from the main component between iterations i_{j-1}^A and i_j^A for $j \geq 1$. The first adaptation point, i.e. the first time q_3 is estimated, is i_A^1. At this point α_3 is switched from $\alpha_3^0 = 0$ to $\alpha_3^1 = 0.5$. This is illustrated in figure 1 where two adaptation schemes are pictured. The adaptation points are marked as dots. In the relatively intensive adaptation scheme adaptation is started early, adaptation is frequent and α_3 increases quickly. In the less intensive scheme more trust is placed in the exploring capabilities of q_1 and q_2 and i_A^1 is therefore larger. Note the relative sense in which we use the term 'intensive'; since both schedules displayed here are much less intensive than those used for the lower dimensional models in Kohn and Giordani (2008). In the evaluation in section 5 the intensive scheme is used with a medium-scale (number of parameters, $n_\theta = 23$) DSGE model and the less intensive scheme with a large-scale ($n_\theta = 51$) model.

Figure 1 Adaptation and hybrid schedules.



In our experience, with a high-dimensional DSGE model the danger of an intensive adaptation scheme is that q_3 fitted on too few draws may provide a poor approximation to the posterior. If, in addition, the share of draws α_3 is increasing steeply early in the chain the overall acceptance rate can be low and in the worst case the sampler collapses.

3.1 ARWM and AUIMH components

The adaptive Metropolis algorithm was introduced by Haario, Saksman and Tamminen (2001). The proposal distribution for the ARWM used in this paper is

$$q_{1,i+1}(\theta|\Theta_i) = \alpha\beta N\left(\theta_i, \kappa_1\Sigma_{i+1}\right) + (1-\alpha)\beta N\left(\theta_i, \kappa_2\Sigma_{i+1}\right) + (1-\beta)N\left(\theta_i, \kappa_3 I\right),$$

where Σ_i is updated using the recursive formulas

$$\Sigma_{i+1} = \frac{i}{i+i_0}\left(\frac{i-1}{i}\Sigma_i + \frac{(i\bar{\theta}_{i-1}\bar{\theta}_{i-1}^T - (i+1)\bar{\theta}_i\bar{\theta}_i^T + \theta_i\theta_i^T)}{i}\right) + \frac{i_0}{i+i_0}\Sigma_0 \tag{8}$$

and

$$\bar{\theta}_i = \frac{i+1}{i_0+i+1}\left(\frac{i}{i+1}\bar{\theta}_{i-1} + \frac{1}{i+1}\theta_i\right) + \frac{i_0}{i_0+i+1}\bar{\theta}_0 \tag{9}$$

such that i_0 is the 'prior sample size' placed on an initial mean $\bar{\theta}_0$ and covariance matrix Σ_0. Here we will use $\bar{\theta}_0 = \theta_m$, where θ_m is the posterior mode and $\Sigma_0 = -H_m^{-1}$ as the initial approximative covariance matrix. The scaling parameters are chosen as $\kappa_1 = 2.38^2/d$, $\kappa_2 = 9\kappa_1$ and $\kappa_3 = 0.1^2/d$ where d is the dimension of θ. The purpose of the second component is to be able to escape local modes (see Silva, Kohn, Giordani and Mun (2008)). The scaling parameter κ_1 is optimal when the target density is Gaussian (Roberts and Rosenthal (2001)). When $\alpha = \beta = 1$ and $i_0 = \infty$ the (non-adaptive) RWM sampler is obtained as a special case. In the adaptive case we let $i_0 = 100$ and $\alpha = 0.95$ in our experiments below.

The unimodal independence proposal is given by the multivariate t distribution

$$q_{2,i+1}(\theta) = t\left(\bar{\theta}_i, \Sigma_{i+1}, v\right)$$

and this component is referred to as adaptive unimodal independence Metropolis-Hastings (AUIMH). The mean and covariance are updated using [8] and [9] and the degrees of freedom, v, is fixed. In our experiments below we let $v = 10$ and $i_0 = 1000$ for the AUIMH sampler. When q_2 is a component in a hybrid sampler we let $\nu = \infty$ such that

$$q_{2,i+1}(\theta) = N\left(\bar{\theta}_i, \Sigma_{i+1}\right),$$

since the presence of fat tails in the proposal distribution (see Kohn and Giordani (2008) for a discussion) appears to be less important for a hybrid sampler. The values chosen for ν are based on a limited amount of experimentation, comparing $v = 5, 10$ and ∞.

3.2 T-copula with marginal mixture of normal component (TC)

The main component of the proposal distribution, q_3, is the t copula with mixture of normal marginal distributions introduced by Silva, Kohn, Giordani and Mun (2008).

Let $t_{d,v}(x|\mu, \Sigma)$ denote the d-dimensional t density function with mean μ, degrees of freedom v and covariance matrix $\tilde{\Sigma} = \frac{v}{v-2}\Sigma$ and let $T_{d,v}(x|\mu, \Sigma)$ denote the corresponding cumulative distribution function. Further, let $f_j(\theta^j|\lambda_j)$ and $F_j(\theta^j|\lambda_j)$ denote the density and distribution function respectively of a univariate mixture of normal where $\lambda_j = (m_j^T \; v_j^T \; p_j^T)^T$ collects the parameters describing the mixture, namely the component means, variances and probabilities. The main component of the proposal density is then described by the mixture

$$q_{3,i+1}(\theta|\Theta_i) = \tilde{\beta} q_{31,i+1}(\theta|\Theta_i) + (1 - \tilde{\beta}) q_{32,i+1}(\theta|\Theta_i)$$

where

$$q_{31} = \frac{t_{d,v}(x|\mu, \Sigma)}{\Pi_{i=1}^d t_{1,v}(x^j|0, 1)} \Pi_{i=1}^d f_j(\theta^j|\lambda_j) \tag{10}$$

and where x^j and θ^j are related by

$$T_{1,v}(x^j|0, 1) = F_j(\theta^j|\lambda_j) \text{ for } j = 1, ..., d \tag{11}$$

The second component q_{32} is a fat-tailed version of q_{31} obtained by inflating the univariate mixture of normal variances by a factor 9. The fraction of draws from q_{32} is set to $1-\tilde{\beta} = 0.05$. We next describe how to (i) estimate q_{31} based on a set of draws $(\theta_0, ..., \theta_{i-1})$, (ii) how to evaluate q_3 and (iii) how to generate a sample from q_3.

3.2.1 Estimation and evaluation of $q_{3,1}$

The mixture of normal marginal densities, f_j, are estimated using clustering based on a sample $\theta_0^j, ...\theta_{\tilde{R}}^j$ where $\tilde{R}$ is the number of draws obtained from the posterior distribution at the point of (re-)estimation, as described in Kohn and Giordani (2008) and Silva, Kohn, Giordani and Mun (2008). The number of mixture components, n_{λ_j}, is selected based on the Bayesian information criterion (BIC), where the maximum number of components is set to 4. An additional step is added here in comparison with Silva, Kohn, Giordani and Mun (2008). In an attempt to improve the fit of the normal mixture its likelihood function is maximised subject to some restrictions: we condition on the number of components n_{λ_j} selected by the BIC criterion and restrict the means, m_j^T, and variances, v_j^T, to some interval around the clustering estimates, $\tilde{m}_j^T$ and $\tilde{v}_j^T$. The probabilities p_j^T are restricted from below.[3] The clustering estimate is used as the starting value for the optimisation routine and a maximum number of mixture of normal density evaluations, e.g. $1,000$, is specified. The modified estimate obtained is denoted $\hat{\lambda}_j$ and it is ensured that $L(\hat{\lambda}_j) \geq L(\tilde{\lambda}_j)$ where L is the likelihood function. The procedure is computationally cheap and works well for simulated iid data. In section 5.4 the procedure is assessed directly in an MCMC sampling context.

The estimated MN densities $f_j(\theta^j|\hat{\lambda}_j)$ and the degrees of freedom v of the t copula are then used to obtain latent variables

$$x^{ji} = T_v^{-1}(F_j(\theta^{ji}|\hat{\lambda}_j))$$

for $j = 1, ..., d$ and $i = 1, ..., \tilde{R}$. By the integral transform x^{ji} has a univariate t distribution $t(0,1,v)$.

Next the latent variables x^{ji} are used to obtain an estimate of the covariance matrix $\hat{\Sigma}$. Finally, the degrees of freedom, v, is updated by selecting the v which maximises [10] on a grid $v \in \{3, 5, 10, 1000\}$, where $v = 1000$ represents a Gaussian copula. The evaluation of the density q_{31} can be inferred from the steps above. The details are given in Silva, Kohn, Giordani and Mun (2008).

3.2.2 Generating a sample

To generate a sample from [10] we proceed as follows. First a vector $x = (x^1, ..., x^d)^T$ is drawn from the multivariate t distribution with mean 0, covariance $\tilde{\Sigma}$ and degrees of

[3] The restrictions are meant to prevent degenerate components with near-zero variances that are likely to otherwise arise in a MCMC setting (Kohn and Giordani (2008)).

freedom v. Each latent variable x^j is transformed into a mixture of normal draw θ^j via

$$\theta^j = F_j^{-1}(T_{1,v}(x^j|0,1))$$

which can be achieved by numerically solving

$$T_{1,v}(x^j|0,1) = F_j(\theta^j|\lambda_j)$$

either using a nonlinear equation solver, e.g. Newton-Rhapson, or by tabulating the implicit relationship between x^j and θ^j on a grid for each parameter $j = 1, ..., n_\theta$. In our implementation of the algorithm we use the former approach.

3.3 Sampler acronyms

In this subsection the sampler acronyms used in section 4 are briefly explained. The random walk Metropolis algorithm is denoted RWM and the $t(v)$ independence sampler is denoted UIMH. The ARWM and AUIMH are the adaptive versions where [8] and [9] are used to update the covariance and mean of the proposal distribution. The ARWM3C algorithm is characterised by $i_0 = 0$ in [8] and [9] such that no a priori information on the covariance matrix of the RWM proposal is used. The t and Gaussian copula samplers are TC and GC, respectively. The RWM-UIMH and the adaptive ARWM-AUIMH are the hybrid samplers where the fraction of RWM steps is α_1 and the fraction of $t(v)$ proposals is $\alpha_2 = 1 - \alpha_1$ and these shares are fixed for the entire chain.

The main sampler is denoted ARWM-AUIMH-TC. Here a fraction α_3 of the proposals are generated by the TC component and this share is increasing through the chain as described above. In the tables of results below the overall fraction of draws from the main component for the chain is reported. The first two components each generate a fraction $\alpha_1 = \alpha_2 = (1 - \alpha_3)/2$ of the draws. In the RWM-TC sampler $\alpha_1 = 1 - \alpha_3$ and $\alpha_2 = 0$.

4 Parallel adaptive hybrid MCMC

An important advantage of independence MH samplers over RWM samplers is their suitability for parallel implementation. The purpose of this section is, first, to reiterate this fundamental point and, second, to explain how parallelisability is affected by adaptation and hybridisation, i.e. the introduction of RWM steps into an otherwise independence chain sampler. A good introduction to parallel MCMC, albeit from a different perspective, is provided by Wilkinson (2006).

Parallel independence chain samplers are attractive in most hardware contexts and we provide two examples of environments which are believed to be relevant to economists and econometricians. First, for personal high performance computing (PHPC) environments, characterised by relatively few, say between 4 and 8, processor cores and perhaps limited

network performance. An example is use of Matlab and its parallel functionality (the Parallel Computing Toolbox) on a multicore desktop computer. Second, for massively parallel computing (e.g. $P > 50$ processors) in high performance computing (HPC) environments, characterised by reasonably fast interconnection networks allowing for fast interprocessor communication. For convenience the discussion is restricted to single block Metropolis-Hastings samplers, which are most suitable for parallelisation.

Let K be the number of synchronisation points in a parallel algorithm, i.e. points of communication between processes/processors (the terms are used interchangeably here). We loosely define a limited communication (LC) parallel MCMC algorithm as an algorithm for which the number of posterior draws per synchronisation point, R/K, is 'large', where R is the total number of draws from the posterior. Therefore LC algorithms are algorithms which should display good scalability properties on a large variety of parallel computers, such that hardware issues can largely be abstracted from. The typical measure of scalability, or parallel efficiency, is the *relative speedup*

$$S(P) = \frac{T(1)}{T(P)} \leq P$$

where $T(P)$ is the wall-clock time of running the parallel program on P processors. An alternative measure, conveying the same information, is the *relative efficiency*, $S(P)/P$.

The parallel approach based on regeneration discussed by Brockwell and Kadane (2005) is an example of LC parallel MCMC whereas the 'parallel blocks' approaches of Wilkinson (2006) and Whiley and Wilson (2004) and the prefetching approach of Brockwell (2006) and Strid (2009) are better described as communication intensive.

Two factors determine whether an adaptive MCMC algorithm can be implemented as a limited communication parallel algorithm: the proposal density and the intensity of adaptation. Consider first the form of the proposal density in the non-adaptive Metropolis-Hastings case. We can distinguish two main classes of algorithms: independence samplers and samplers where the proposal density depends on the current state of the chain, e.g. the RWM sampler. For a single-block independence sampler a trivial two-stage parallel algorithm can be constructed. In the first stage, proposals for the complete chain are evaluated in parallel. Using a homogeneous parallel computer, i.e. one where all processors have the same performance, each process generates and evaluates R/P proposed parameters and a master process collects the results. In the second stage the MH algorithm is run serially by the master process with the posterior values at the proposed parameters available from the first stage. This parallel independence MH algorithm displays extremely good scalability. Furthermore parallel efficiency is largely independent of the properties of the parallel computer since there is only one synchronisation point in the algorithm. It is also easy to adapt the algorithm to the less relevant case of inhomogeneous processors.

Employing $P = 64$ processors on a standard HPC cluster to estimate a medium-scale DSGE model with this simple algorithm and a $t(v = 10)$ proposal distribution, Strid (2009) reports the speedup $S(64) = 63$. The unreliable statistical performance of the

independence sampler in that example, however, implies that it is an extremely wasteful approach. We return to this model below.

The single-block RWM or ARWM algorithm has $K = R$ such that limited communication parallel implementation is precluded. Parallel prefetching RWM or ARWM algorithms can be implemented with $K < R$ but these implementations will remain communication intensive. Using $P = 8$ processors for a close to optimal prefetching RWM algorithm Strid (2009) reports $R/K \approx 5$. Prefetching algorithms can be expected to perform well on clusters with a fast interconnection network and when few, maximally $P = 10 - 15$, processors, are employed.

In the adaptive sampling framework the frequency of adaptation, i.e. how frequently the proposal density is updated, is also of importance. Naturally, an adaptation point implies a synchronisation point, such that $M \leq K$ where M is the number of adaptation points. If adaptation is performed at each iteration ('online') we have $K = M = R$. The 'pure', i.e. not hybrid, AIMH sampler with batch estimation of the proposal density is characterised by $K = M << R$ where M is the total number of adaptation points. Kohn and Giordani (2008) suggest $R/M = 1000$ in their implementations, abstracting from a short initial phase where updating of the proposal is more frequent. It is immediately clear that a parallel version of their batch updating AIMH algorithm will display very good parallel efficiency. Note, however, that the adaptation schedule was motivated solely on statistical grounds rather than being based on parallel efficiency concerns.

Three basic requirements for limited communication single-chain parallel adaptive MCMC algorithms follow. First, the majority of the draws must be based on independently generated proposals. Random walk steps may be used but parallel efficiency will suffer. Second, adaptation can not be too frequent. In the context of adaptive independent samplers with batch updating this does not appear to be restrictive. Third, deterministic hybrid schemes must be used rather than random mixture hybrids in order to avoid randomly distributed synchronisation points in the algorithm.

The key trade-off in our framework is related to hybridisation, i.e. the insertion of random walk steps. In practise we expect the random walk steps to be much more frequent than the adaptation points and hence the number of synchronisation points, K, will coincide with the number of random walk steps. From a statistical perspective these are expected to make the sampler more robust, reliable and efficient in medium and high dimensional settings. From the parallel computing perspective a random walk step implies a synchronisation point, i.e. interprocessor communication, which is costly.

Based on these considerations we construct and attractive simple benchmark parallel hybrid single-block MCMC algorithm. The algorithm combines draws from an independence proposal which approximates the posterior well on most of its support, e.g. the t copula used in this paper, with occasional RWM or 'always reject' prefetching RWM steps in a deterministic cycling scheme. The prefetching RWM step is simply the parallel analogue of the RWM step. With P processors, in the 'always reject' variant of prefetching

the posterior kernel is evaluated in parallel at the points

$$\theta^p = \theta_i + \varepsilon^p, \ p = 1, ..., P, \tag{12}$$

where θ_i is the current state of the chain. Intuitively the 'always reject' prefetching RWM step either allow us to escape a 'difficult point', which is desirable from a Markov chain efficiency perspective, or it does not. In the latter event the prefetching step will 'at least' be computationally efficient, since all P posterior evaluations generate draws.

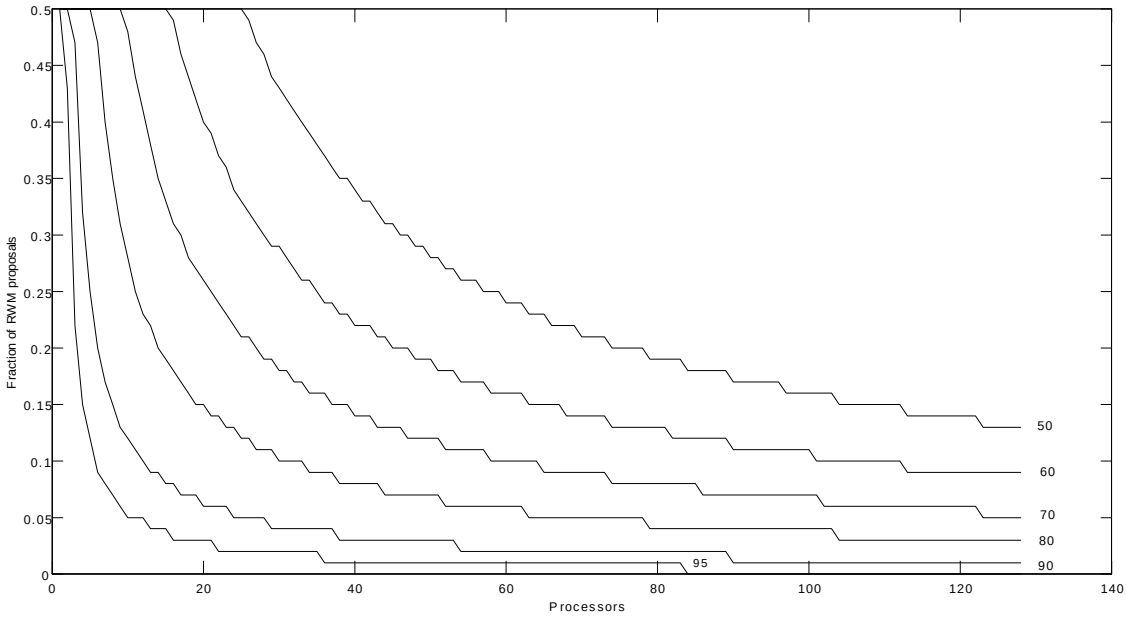
Ultimately overall efficiency (i.e. combined Markov chain statistical efficiency and parallel computational efficiency) as measured by iid equivalent draws per time unit, hinges mostly on the quality of the independence proposal. A good independence proposal increases Markov chain efficiency and therefore allows for infrequent RWM steps, which in turn boosts parallel efficiency by reducing the number of synchronisation points K in the algorithm. To summarise: a key feature of this parallel adaptive and hybrid algorithm is the *complementarity of statistical and parallel efficiency*. This may be contrasted with the 'always reject' RWM prefetching algorithm where increased parallel efficiency is traded against lower Markov chain efficiency, i.e. the optimal scaling factor, κ_1, is smaller with $P > 1$ processors than with $P = 1$ processor (Strid (2009)). Pseudo-code for the algorithm is given in the appendix.

The practical approach we suggest with the parallel hybrid algorithm is to select the fraction of RWM steps (α_1 in the notation of section 3) to achieve good statistical performance and then let the parallel performance be determined residually. Once a suitable fraction of draws from the main component, α_3, has been established on statistical grounds it would presumably be possible to increase overall efficiency by increasing α_3 (and hence decreasing α_1) further, thus increasing parallel efficiency but reducing Markov chain efficiency. However, we do not expect this type of tuning to be worthwhile in practice.

In figure 2 theoretical relative efficiency contours, $S(P)/P$, of the algorithm is plotted as a function of the fraction of RWM proposals, α_1, and the number of processors, P. The assumptions and calculations underlying the figure are described in the appendix. An example: if $\alpha_1 = 10\%$ and $P = 64$ the theoretical relative efficiency is roughly 70%, implying that the theoretical speedup is $S(64) = 45$. In practise the speedup would be lower but the numbers are still indicative of the enormous computational gains associated with 'largely independence' samplers.

Finally we note that the adaptation step, i.e. the mixture of normals estimation, is easy to parallelise since it is performed independently for each parameter θ^j. In general $\text{mod}(n_\theta, P) \neq 0$ implying that, on a homogeneous parallel computer, each process performs adaptation for $floor(n_\theta/P)$ or $ceil(n_\theta/P)$ parameters. Possible implications are that we may want to perform adaptation more frequently and/or spend more resources on obtaining a good proposal distribution.

Figure 2 Relative efficiency contours, $S(P)/P$, for the parallel adaptive and hybrid Metropolis-Hastings algorithm (in percent).



5 Algorithm comparison

5.1 A Smets and Wouters type model

The statistical performance of the sampling algorithms is first compared using one of the core macroeconomic models at the European Central Bank, a slightly simplified version of the Smets and Wouters (SW) model (Smets and Wouters (2003)). The nominal frictions in the model are sticky prices, sticky wages and money demand by both households and firms. The real frictions are monopolistic competition in the markets for intermediate goods and labor, internal habit formation in preferences for consumption, investment adjustment costs and capital utilisation costs. In our version of the model there are $n_\epsilon = 8$ structural shocks: consumption preference, labor supply, stationary technology, unit root technology, government spending, monetary policy, inflation target and investment adjustment cost shocks. Similar models have been analysed and/or estimated in many studies (Smets and Wouters (2003); del Negro, Schorfheide, Smets and Wouters (2005)).

The number of state variables in the model is $n_x = 15$ and $n_\theta = 23$ parameters are estimated (as customary a subset of the structural parameters are calibrated, e.g. parameters which are not well identified by the data). The model is estimated on Euro Area data for the period 1982Q1-2003Q4 (88 observations). The $n_y = 5$ data series used

for estimation are the short-term interest rate, inflation, output growth, the consumption-to-output ratio and the investment-to-output ratio.

The model is solved using the method of Klein (2000) and the Kalman filter is initialised using the stationary distribution of the state variables, which is obtained by solving the Lyapunov equation, [4], numerically. The prior distribution is 'standard' in two senses: first, parameters which are unbounded, bounded from below and bounded both from below and above have normal, (inverse) gamma and beta prior distributions, respectively. Second, we have attempted to select hyperparameters for these distributions which are similar to those used in previous studies based on similar models, e.g. the articles mentioned above.

Chains are obtained for (at least) two initial values per algorithm, the (presumed) posterior mode, θ_m, and the prior mode, θ_{prior}. The log posterior is optimised using simulated annealing (SA), with the set of optimisation parameters 'conservatively' chosen such that the mode is slowly approached, and using Christopher Sims' optimiser *csminwel* for a variety of starting values drawn from the prior distribution (Sims (1999)). Both approaches appear to work well. The log posterior kernel (and likelihood) values at θ_m and θ_{prior} are 1754.32 (1768.15) and 1630.69 (1632.06) respectively. The initial value for the chain does not influence the algorithmic comparison and therefore results are provided for the chains initialised at the prior mode. For each algorithm $R = 500,000$ draws are obtained and the first $100,000$ are discarded as burn-in.

The empirical optimal acceptance rate for the standard RWM algorithm is in the range $0.20 - 0.24$, based on the average inefficiency factors from 10 chains with acceptance rates in the interval $0.02 - 0.40$. In table 1 the inefficiency factors for the algorithms are presented. The inefficiency factors are given by

$$\tau_j = 1 + 2\sum_{k=1}^{K} Corr\left(\theta_i^j, \theta_{i+k}^j\right), j = 1, ..., n_\theta, \tag{13}$$

where $K = 500$. This measure indicates how many draws are needed using a given algorithm to achieve the same numerical efficiency as when (hypothetically) sampling an independent draw from the posterior distribution.

The main observations are as follows. First, all samplers obtain the same posterior distribution. In the case of the UIMH sampler the inefficiency factors differ in repeated runs and only the best result, i.e. the run which produced the lowest mean inefficiency, is reported here. This is due to the randomness in the occurrence of difficult points, see also the discussion in Adolfson, Lindé and Villani (2007*a*). Second the mean (or median) inefficiency for the main sampler, the ARWM-AIMH-TC hybrid algorithm, is roughly a factor $6 - 7$ lower than for the RWM algorithm. Note again that the model parameters have been transformed in the 'standard' way, described in section 2. This improves the performance of the RWM algorithm. In the original parameterisation the difference in inefficiency factors would presumably be larger. Third, the choice of initial sampler for

Table 1 Inefficiency factors and acceptance rates for the SW type model

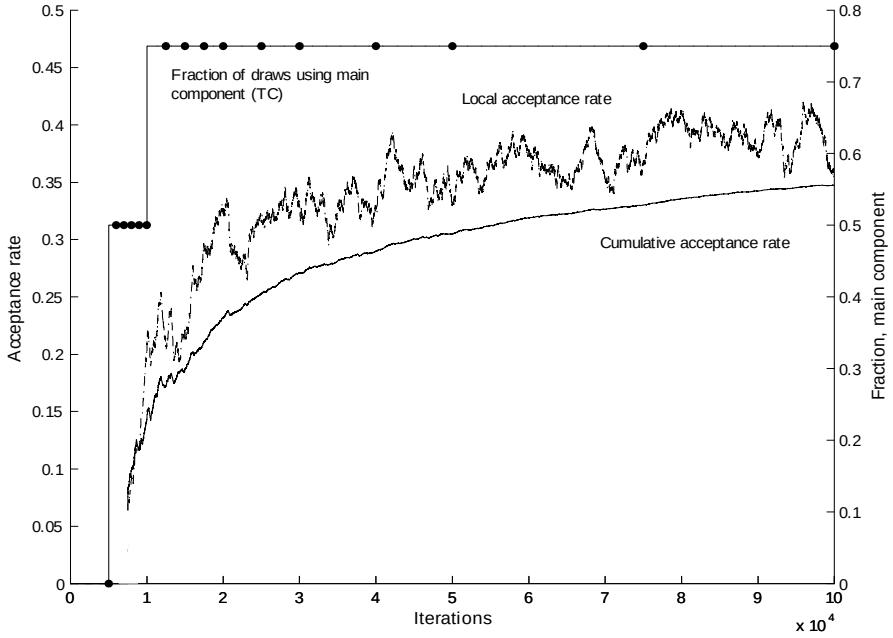
Algorithm	Accept. rate (%)	Inefficiency Min	Median	Mean	Max	Overall fraction main comp. (%)	Max no. of clusters
RWM	24.2	66.0	85.4	99.2	163.9	-	-
UIMH (t, df=10)	17.0	62.6	104.8	141.3	410.1	-	-
ARWM3C	24.0	83.3	104.0	105.7	140.6	-	-
RWM-UIMH	24.6	31.2	48.6	64.1	126.1	50 (UIMH)	-
ARWM-AUIMH	21.8	20.2	22.8	23.9	32.3	50 (AIMH)	-
RWM-TC	37.7	9.2	14.6	15.0	36.8	74.0 (TC)	4
RWM-UIMH-TC	44.5	8.2	15.2	17.0	57.7	95.6 (TC)	4
ARWM-AUIMH-TC	44.0	8.7	12.9	14.1	40.6	95.6 (TC)	4

the hybrid algorithm does not appear crucial. The ARWM-AIMH initial sampler delivers a better approximation q_3 quicker but the simpler RWM initial sampler yields similar inefficiency factors. The adaptation and hybrid schedules used with the RWM-UIMH-TC and ARWM-AUIMH-TC samplers are displayed in figure 1. Third, the simple ARWM-AIMH sampler reduces inefficiency factors by roughly a factor 4 in comparison with the RWM algorithm.

In figure 3 the performance of the hybrid RWM-TC sampler is illustrated graphically using the first $100,000$ draws from the sampler, corresponding to the burn-in phase. The acceptance rate of the main component is displayed along with the fraction of draws from the main component. The adaptation points are marked as dots on the hybrid schedule. The acceptance rate is also calculated for a moving window of $2,000$ draws to provide an idea of how the quality of the main component improves, a 'local' acceptance rate. Adaptation starts after $i_A^1 = 5,000$ draws and is thereafter frequent. The main component generates 50% of the draws up to $i_A^6 = 10,000$ and after that point it accounts for 75% of the proposals. At the end of the run 74% of the proposals have been generated by the TC component. The acceptance rate improves quickly during the intensive adaptation phase and after roughly $30,000$ draws it appears that a good proposal has been found. The acceptance rate of the TC component is initially below 10% and after $30,000$ iterations it has reached roughly 30%. The quality of the posterior approximation improves at a slower rate subsequently and adaptation becomes less frequent. Eventually the acceptance rate of the main component settles down at 44% which can be compared to the 17% acceptance rate obtained with a standard independence sampler which uses a t proposal (UIMH).

The final proposal distribution for the samplers with a TC component, which is constructed at iteration $150,000$, has more than one component in the mixture of normal marginal density for 5 of the 23 estimated parameters. The degrees of freedom estimate for the copula in [10] is $v = 1000$ for the samplers with a TC component, i.e. a Gaussian copula is selected by the adaptive procedure. Four of these five parameters are $AR(1)$ coefficients of shock processes, with beta prior distributions on the interval

Figure 3 Performance of the RWM-TC sampler for the SW type model.

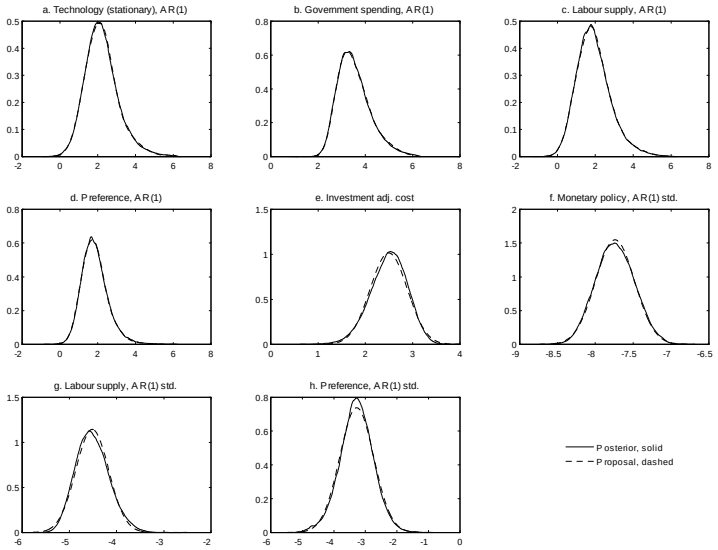


$[0, 1]$. The marginal posterior density (Epanechnikov kernel smoothing density estimate) and the final mixture of normal density estimate for these parameters are displayed in figure 4, panels a-d (in the transformed parameter space). The marginal proposals appear to provide a good fit to the marginal posteriors, which are skewed to the right for these parameters. Intuitively, the improved ability to fit these skewed parameters explain the difference in sampling performance between the TC sampler and the samplers with (A)UIMH components. In panels e-h the proposal and posterior densities are shown for the four parameters which, by visual inspection, display the largest discrepancy between the marginal proposal and posterior densities. For these parameters the final marginal proposal densities are normal distributions, i.e. the BIC criterion selects one component MN densities. The marginal posterior distributions of the remaining parameters, those which are not displayed here, are symmetric in shape and apparently well approximated by a normal distribution.

5.2 The RAMSES model

The RAMSES model is a large-scale DSGE model developed and estimated by researchers at the Swedish central bank (Adolfson, Laséen, Lindé and Villani (2007); Adolfson, Lindé and Villani (2007*b*); Adolfson, Laséen, Lindé and Villani (2008*a*); Adolfson, Laséen, Lindé

Figure 4 Marginal posterior and mixture of normal proposal distributions, SW type model.



and Villani (2008*b*)). It is essentially the extension of the SW model to a small open economy setting. A number of interesting econometric issues related to the estimation of the model are discussed in Adolfson, Lindé and Villani (2007*a*). The analysis here is restricted to a comparison of sampling performance and the reader is referred to the mentioned papers for a detailed discussion of the content of the economic model, variations of the model, the data used for estimation and the choice of prior distribution.

The LGSS model contains $n_x = 60$ state variables and $n_\epsilon = 21$ shocks. The model is estimated using Euro Area data for the period $1970Q1 - 2002Q4$ and $n_y = 15$ observed series are used to estimate $n_\theta = 51$ of the model's structural and auxiliary parameters. Again a number of parameters are calibrated. In Adolfson, Laséen, Lindé and Villani (2007) the model is estimated with and without variable capital utilisation. Here only the former case is considered.

The model is solved using the AIM algorithm. The initial, $1970Q1$, state distribution for the Kalman filter is diffuse. The observations for the $1970s$ are only used to obtain a prior for the state variables in $1979Q4$ and inference is based on data for the period $1980Q1-2002Q4$. The prior distribution is standard in the sense described in the previous section.

For each algorithm $1,000,000$ draws are obtained and every 5^{th} is kept for inference. The burn-in is $200,000$ draws. Inefficiency factors and acceptance rates for the various samplers are presented in table 2, calculated based on the thinned chains and with $K = 1,000$ in [13]. The performance ranking of the algorithms resemble the one for the

Table 2 Inefficiency factors and acceptance rates for the RAMSES model

Algorithm	Accept. rate (%)	Inefficiency Min	Median	Mean	Max	Overall fraction main comp. (%)	Max clusters
RWM	22.7	27.4	41.2	67.2	748.4	-	
IMH (t, df=10)	8.0	125.8	455.6	459.1	948.2	-	
RWM-UIMH	16.7	26.9	41.4	92.3	830.5	50%(IMH)	
ARWM-AUIMH	16.1	13.0	15.6	22.4	135.3	50%(AIMH)	
RWM-TC	19.8	7.9	11.2	14.2	49.5	77.5%(TC)	4
ARWM-AIMH-TC	23.5	8.3	12.8	14.8	33.7	77.5%(TC)	4

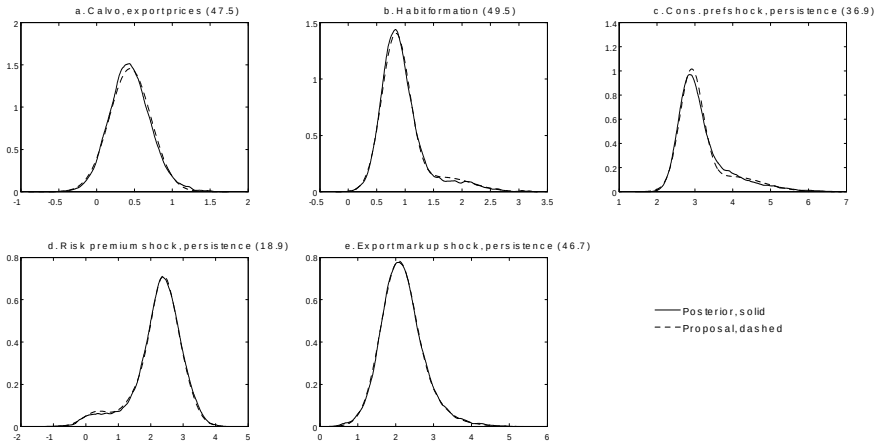
smaller DSGE model above. All samplers find the same, and therefore presumably correct, posterior distribution (with the IMH algorithm possibly being an exception). The median or mean inefficiency factor is roughly $3-4$ times lower for the samplers with a TC component when compared to the benchmark RWM algorithm.

The final proposal distributions of the RWM-TC and ARWM-AIMH-TC samplers, constructed at iteration $500,000$, have more than one component in the mixture of normal density for 8 out of the 51 parameters and, notably, 7 of those 8 parameters have beta prior distributions. In these instances the marginal posterior distributions are apparently skewed, even after reparameterisation. Furthermore, among the 25 parameters with beta prior distributions these 7 are among those with the most skewed prior, e.g. shocks which are a priori thought to be highly persistent. These eight multi-component, or 'problematic', parameters also correspond to those with the largest chain autocorrelations when the RWM algorithm is used. The median inefficiency factor for this subgroup of parameters is 180, which can be compared to the overall median inefficiency factor of 41 presented in the table. For the RWM-TC sampler the median inefficiency for the group is 24, to be compared with the overall median of 11.2.

In figure 5 the marginal posterior (again the Epanechnikov kernel smoothing density estimate) and the final RWM-TC marginal proposal distributions for the five parameters with the largest RWM-TC sampler inefficiency factors are displayed (with the inefficiency factors, τ_j, in parenthesis). Note that these are also the parameters with the largest RWM inefficiency factors. Clearly, the marginal posterior densities with prolonged tails in panels b, c and d are not possible to approximate well with a symmetric proposal density. The reduction in inefficiency factors when using the TC-component samplers is most dramatic for these parameters.

These parameters also display irregular posterior dependencies, which may be illustrated via bivariate posterior plots. The correlation between the export parameters - the Calvo export price stickiness (ξ_x, in the notation of Adolfson, Laséen, Lindé and Villani (2007)) and the export markup shock persistence (ρ_{λ^x}) parameters - is -0.66 and the correlation between the consumption parameters - the habit formation (b) and preference shock (ρ_{ζ^c}) parameters - is -0.77. In both instances these correlations can be intuitively

Figure 5 Marginal posterior and mixture of normal proposal distributions for the five parameters with largest inefficiency factors, RAMSES model.



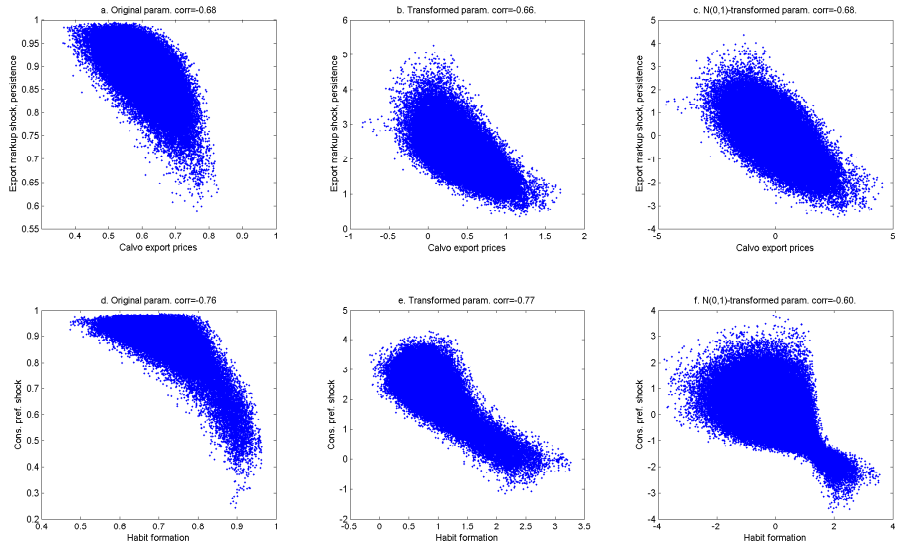
explained, e.g. concerning the export parameters the lack of appropriate export price data explains the weak identification of these parameters (Adolfson, Laséen, Lindé and Villani (2008*a*), p. 8).

In figure 6 bivariate plots of the draws from the RWM-TC sampler for these two pairs of parameters are displayed. In the left part of the figure, panels a and d, the parameters are displayed in the original (bounded) parameterisation and in the middle panels, panels b and e, in the standard (unbounded) transformation described previously. For both pairs of parameters reparameterisation straightens out the banana-shaped bivariate posterior surface somewhat but the shape is still non-elliptical.[4]

In the final proposal distribution the degrees of freedom estimate for the copula in [10] is, as for the SW type model above, $v = 1000$ for the samplers with a TC component, i.e. a Gaussian copula is selected endogenously by the adaptive procedure. In the right part of the figure, panels c and f, the parameter draws have been transformed to $N(0, 1)$-variables via the relation [11]. The mixture of normal densities used for these transformations are estimated using all the $160,000$ post burn-in draws obtained from the RWM-TC sampler. The racquet shape of the transformed bivariate posterior in panel f is suggestive of the limits as to what can be achieved with a Gaussian copula approximation in this case. Although the marginal posteriors in figure 5 appear reasonably well approximated by mixture of normals densities their joint dependence does not appear well captured by the simple Gaussian copula.

[4]The mean inefficiency factors of the RWM sampler reported by Adolfson, Lindé and Villani (2007*a*) based on $500,000$ draws (and no thinning) are 339.25 and 217.13 in the original and transformed parameter spaces, respectively.

Figure 6 Bivariate posterior plots for two pairs of parameters, RAMSES model.



It is not surprising that estimation of the RAMSES presents more of a challenge than estimation of the SW type model in the previous section, given the higher dimensionality of the parameter vector of estimated parameters. Figures 4 and 5, which display the subset of the most problematic parameters for each model, also suggest that the posterior distribution of the RAMSES model is further away from normality than the posterior of the smaller SW model.

In our experience, with the SW type model even a poorly designed adaptation schedule, i.e. too intensive given the number of parameters, works well, in the sense that the sampler is able to recover from a bad start and eventually obtains the posterior distribution. For the RAMSES model it is more important to choose a slower intensity of the adaptation scheme: if adaptation is too intensive the sampler will not be able to recover from an initial poor approximation of the posterior.

5.3 Nonlinear estimation of a small-scale DSGE model

We now apply the sampling algorithms to the estimation of a small-scale second order approximated DSGE model. The likelihood function is evaluated using a standard particle filter, see e.g. Arulampalan, Maskell, Gordon and Clapp (2002). The computational cost of particle filtering implies that the models that are considered for nonlinear estimation in practise are small. This, in turn, implies that the chances of success for a one-block MCMC approach increases since the dimension of the parameter vector, n_θ, is smaller.

The prototypical small New Keynesian (NK) model is borrowed from An (2005). The

state-space representation of the model is given by the state equation, [5] and [6], and the observation equation [3], which are obtained by first solving the model using the approach of Schmitt-Grohe and Uribe (2004). The model contains $n_{x_1} = 3$ shocks, a total of $n_x = 7$ state variables and $n_y = 3$ observed variables.

Since the main focus here is on comparing sampling performance we choose to use simulated data in order to abstract, as much as possible, from issues related to particle filtering performance. A bad model fit or the presence of outliers in the data, for example, can have a severe impact on the performance of the filter, which would prevent a meaningful comparison of samplers. The construction of improved particle filters to be able to deal with such issues, although a very important topic, is beyond the scope of this article.

The data-generating process (DGP), i.e. the calibration of the model parameters, and the prior distribution for $n_\theta = 13$ estimated parameters is chosen largely as in An (2005). Data for $n_y = 3$ series - output growth, inflation and the nominal interest rate - is simulated from the nonlinear state-space model for $T = 90$ time periods. In the estimation the first 10 periods of the time series are merely used to provide a prior for the state distribution at time $t = 10$ and the remaining 80 periods are used for inference. The model is estimated with the variance of the measurement errors, $\Sigma_v = diag(\Sigma_v)$, calibrated at values which imply that roughly $15 - 20\%$ of the variance of the observed series is due to these errors. (Note that the variance of the measurement errors affects the performance of the particle filter.) The data set generated by the DGP and the prior distribution imply a unimodal posterior distribution.

The key structural parameters for generating nonlinearity in the model are, first, the inverse elasticity of intertemporal substitution, τ, (which determines the curvature of the household's utility function) and, second, the inflation response of the monetary authority, ψ_1. The degree of nonlinearity increases as τ increases and as ψ_1 approaches (from above) the boundary of the indeterminacy region, $\tilde{\psi}_1 = 1$. In the DGP these parameters are assigned the values $\tau = 2$ and $\psi_1 = 1.5$, which implies a 'mildly nonlinear' parameterisation. By experimentation we have found the remaining parameters to be largely unimportant in producing any substantial differences between the linear and nonlinear state-space models.[5]

In the nonlinear case, given that the PF yields only an approximation to the likelihood, it is not possible to obtain the exact mode of the posterior density. The standard approach in the linear case, i.e. to use the negative inverse Hessian evaluated at the posterior mode as the covariance of a RWM proposal, is therefore not available. The RWM sampler is initialized instead using the estimated covariance matrix from a preliminary run, a form of preliminary adaptation, and the chosen scaling factor implies an acceptance rate of 24.0% when the number of particles is $N = 40,000$. In the preliminary run the scaled

[5] By a 'nonlinear parameterisation' θ_{dgp} we, loosely speaking, mean a parameterisation for which data generated from the linear and quadratic state-space models, respectively, display different properties, e.g. the second moments of the data display clear differences.

negative inverse Hessian at the posterior mode of the corresponding log-linear model is used as the covariance of the RWM proposal.

The inefficiency factors of three samplers, RWM, ARWM-GC and ARWM-AUIMH-GC, are presented, for chains started at the prior mode. In preliminary runs with short chains it is found that the TC copula selects the degrees of freedom $v = 1000$ and in the reported chains we fix this value, i.e. a Gaussian copula is pre-imposed. In each case $R = 150,000$ draws from the posterior distribution are obtained and the first $20,000$ are discarded as burn-in. Pure independence samplers, UIMH (t with $v = 10$) and GC, were trapped at 'difficult points' for spells long enough to conclude that these samplers can not be applied successfully to this problem.

The relatively small number of parameters and, more importantly, the computational burden associated with particle filtering suggests a more intensive adaptation scheme, in comparison with the linear estimation examples above. Here adaptation begins at iteration $i_1^A = 2,000$ and is performed every 500^{th} iteration up to $i_{17}^A = 10,000$. After that re-estimation of the MN marginal proposals is performed every $1,000^{th}$ iteration up to $i_{37}^A = 30,000$, from there on every $2,000^{th}$ iteration up to $i_{67}^A = 90,000$ and for the remaining part of sampling every $10,000^{th}$ iteration. Note that even with such frequent adaptation the re-estimation of the proposal density accounts for a negligible share of overall computational time.

In table 3 the inefficiency factors are presented. First the performance of the RWM sampler using $N = 40,000$ and $100,000$ particles in the SPF is compared. The brute force approach of using more particles, in order to obtain a sharper estimate of the likelihood, apparently increases the acceptance rate and decrease the inefficiency factors somewhat. 'Difficult points' now occur *also* as a result of the sampling variability in the likelihood estimator. The standard deviation of the likelihood decreases at rate $\sqrt{N}$, such that increasing the number of particles reduces this problem, but at a high computational cost.

Second, the ARWM-GC and ARWM-AUIMH-GC samplers reduces the inefficiency factors by roughly a factor 4 and $5-6$, respectively, in comparison with the RWM sampler. Two of the 13 parameters have more than one component in the final MN marginal proposal density. The fraction of copula-generated draws, α_3, of these samplers reaches its final level after $20,000$ iterations and for the ARWM-GC sampler a comparison is made for the cases where this fraction is 75% and 90%, respectively. The difference in inefficiency factors is quite small, suggesting that the exact fraction is largely unimportant, at least in a serial computing context (see the discussion below).

Finally, in our serial Matlab implementations of the samplers the time of a particle filter likelihood evaluation for $T = 90$ time periods and $N = 40,000$ particles is $2 - 2.5$ seconds depending on the hardware used such that $R = 150,000$ draws are obtained in roughly $80 - 100$ hours. Fortran mex routines are used for the parts of the particle filter that cannot be written as vectorised code, e.g. the systematic resampling procedure.

Table 3 Inefficiency factors and acceptance rates for the NK model.

Algorithm	Accept. rate(%)	Inefficiency Min	Med.	Mean	Max	Overall frac. main comp.(%)	Max clus.	No. of particles
RWM	24.0	79	104	117	217			40,000
RWM	24.9	75	84	89	113			100,000
ARWM-GC	31.9	12	23	32	86	86.8 (GC)	4	40,000
ARWM-GC	29.7	15	21	28	70	74.0 (GC)	4	40,000
ARWM-AUIMH-GC	30.1	14	17	20	36	74.0 (GC)	4	40,000

These numbers are representative of what can be achieved with reasonably optimised code on a modern desktop computer. The particle filter accounts for more than 98% of estimation time and the time increases linearly in the number of particles, N.

In a serial computing context a comparison of overall, statistical and computational, efficiency then reduces to a comparison of inefficiency factors, i.e.

$$\frac{\bar{\tau}^{rwm}}{\bar{\tau}^{arwm-gc}}\frac{T_{rwm}(1)}{T_{arwm-gc}(1)} \approx \frac{\bar{\tau}^{rwm}}{\bar{\tau}^{arwm-gc}}.$$

where $\bar{\tau}$ is the average inefficiency factor and $T(1)$ is the time of execution of the sampling algorithm on one processor. In a parallel computing context

$$\frac{\bar{\tau}^{rwm}}{\bar{\tau}^{arwm-gc}}\frac{T_{pf-rwm}(P)}{T_{arwm-gc}(P)} \approx \frac{\bar{\tau}^{rwm}}{\bar{\tau}^{arwm-gc}}\frac{S_{arwm-gc}(P)}{S_{pf-rwm}(P)} > \frac{\bar{\tau}^{rwm}}{\bar{\tau}^{arwm-gc}},$$

since $S_{arwm-gc}(P)/S_{pf-rwm}(P) > 1$, where $S_{pf-rwm}(P)$ is the speedup of the prefetching RWM algorithm and $S_{arwm-gc}(P)$ is the speedup of the parallel adaptive and hybrid algorithm outlined in section 4. The speedup ratio increases in P and decreases in the fraction of random walk steps, α_1. Here the speedup is compared with the prefetching (A)RWM algorithm, based on the assumption that it is the best possible way of parallelising the (A)RWM algorithm. In hardware contexts where prefetching is infeasible, due to immense costs of interprocessor communication, the relative attractiveness of the 'largely independence' approach increases, since then $S_{pf-rwm}(P) = 1$.

5.4 Modified estimation of the mixture of normal marginal proposal densities

In this section the modified mixture of normals estimation procedure described in section 3.2.1 is assessed via a simple experiment. The RWM-GC sampler is applied for estimation of the log-linearised version of the small-scale NK model in section 5.3. We generate 100 vectors of starting values from the fattened prior distribution and for each starting vector a pair of chains, each of length $R = 40,000$, are obtained. It is sufficient for the purposes

here to study the behaviour of the GC component only during the critical start-up phase of the chain. For the first chain in a pair the clustering approach is used for estimation of the mixture of normal proposal densities and in the second chain the modified clustering approach is applied. The data set generated by the DGP, the prior distribution and the adaptation schedule are the same as in the previous section, except that adaptation starts already at iteration $1,000$. There are 43 points of adaptation for a chain. The relatively intensive adaptation is desirable since the purpose of the experiment is to isolate the effect of the modified MN estimation procedure on sampling efficiency. The targeted acceptance rate of the RWM component is roughly $24-25\%$.

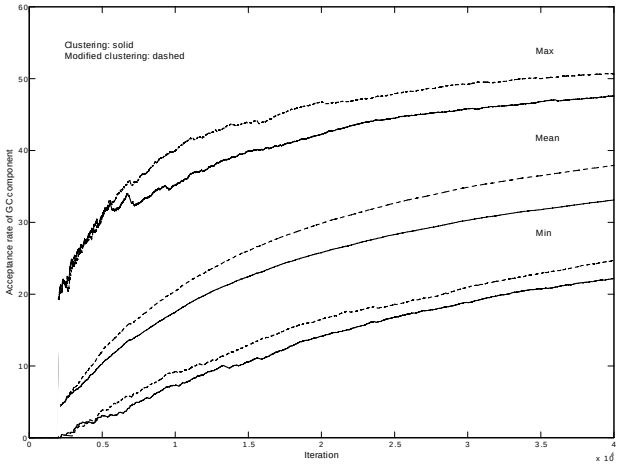
In figure 7 the minimum, mean and maximum (cumulative) acceptance rate of the GC component at each iteration is displayed for the clustering and modified clustering chains, respectively. The statistics are calculated at each iteration based on 100 observed acceptance rates, one per chain. In this stylised experiment the acceptance rates of the chains which use modified clustering estimation is higher. Indirectly this shows that the approximation to the posterior in the early stage of the chain is improved. At iteration $40,000$ the mean of the GC component acceptance rate is roughly 38% for the modified clustering chains and 33% for the clustering chains. For 99 of the 100 pairs of chains the acceptance rate of the modified clustering chain is larger at the final iteration. The number of parameters with more than one component in the MN density is typically 3 or 4 (out of $n_\theta = 13$) at the final iteration.

The experiment also illustrates the robustifying effect of the RWM step. The worst performing chain initially has a GC component acceptance rate very close to zero but yet, as the RWM component continue to generate accepted proposals, it eventually manages to 'take off'.

6 Conclusions

Bayesian inference for DSGE models is typically carried out by single block random walk Metropolis sampling. The high computing costs involved are an obstacle to model development and use, and the poor mixing of RWM in high dimensions further increases computing times and reduces the probability of escaping local modes. This paper combines two features, parallelisation and adaptive independent Metropolis-Hastings, to achieve large computational gains in DSGE estimation. The history of the draws is used to continuously improve a t-copula proposal distribution. A novel twist is an adaptive random walk step inserted at predetermined intervals. Compared to a pure independent sampler, this step improves the ability of the chain to escape difficult points of the type often encountered in medium and high dimensional DSGE models, while maintaining a high statistical efficiency. In linear estimation of a medium scale (23 parameters) and a large scale (51 parameters) DSGE model, the efficiency gains are $6-7$ and $3-4$ times respectively. In a nonlinear estimation example, where computing time is even more of a

Figure 7 Acceptance rate (%) of GC component: min, mean and max for 100 chains, log-linear NK model.



concern, the efficiency gains are $4-6$. The sampler is also better suited to parallelisation than RWM or blocking strategies, so that the effective computational gains can potentially be much larger. Our preferred algorithm employs a copula approach which, based on results by Silva, Kohn, Giordani and Mun (2008), should provide even greater gains over RWM when using non-standard priors such as truncated or mixture distributions.

Acknowledgements

The authors would like to thank Mattias Villani for providing computer code for the RAMSES model and for helpful discussions.

Appendix: Parallel adaptive and hybrid Metropolis-Hastings sampler

In this appendix pseudo-code for a parallel adaptive and hybrid Metropolis-Hastings sampler is presented. Let

$$q_j\left(\theta|\theta^i\right) = q_j\left(\theta\right), \; j = 1, ..., M$$

be a sequence of independence proposal densities and let

$$A = \left\{i_0^A, i_1^A, i_2^A, ..., i_M^A\right\} \tag{14}$$

denote the adaptation points at which these proposal densities are constructed, where $i_0^A = 0$ and where M is the number of adaptation points. Let P denote the number of processes/processors and let R be the desired total number of draws from the sampler. The number of synchronisation points is K.

Algorithm 1 *Parallel adaptive and hybrid Metropolis-Hastings sampler*

1. The current state of the chain is θ_r and the draw counter has the value $D = r$ when the $(r+1)^{th}$ iteration begins and $i_j^A > r \geq i_{j-1}^A$.

2. (Random walk step) Perform a serial RWM step, producing 1 draw, or a parallel 'always reject' RWM prefetching step, producing D_{pf} draws, where $P \geq D_{pf} \geq 1$.

3. (Independence step) Each process p, $p = 1, ..., P$, generates $\theta_p = \left\{\theta_{1p}, \theta_{2p}, ..., \theta_{R_p p}\right\}$ where
$$\theta_{ip} \sim q_j, \ i = 1, ..., R_p,$$
and collects the values of the posterior evaluated at these parameters in the vector $p_p = \left\{p\left(\theta_{1p}\right), p\left(\theta_{2p}\right), ..., p\left(\theta_{R_p p}\right)\right\}$ (parallel).

4. The master process gathers θ_p and p_p, $p = 1, ..., P$ (*gather*).

5. (Metropolis-Hastings step) Run the Metropolis-Hastings algorithm with the posterior already evaluated at $\Sigma_{p=1}^P R_p$ parameter values (serial).

6. Step 1 to 4 produced $D_{pf} + \Sigma_{p=1}^P R_p$ draws. Update the draw counter $D = D + D_{pf} + \Sigma_{p=1}^P R_p$.

7. If $D \geq i_j^A$ perform adaptation of the independence proposal density f and let $j = j + 1$. Otherwise return to 1. (serial or parallel)

8. Return to 1. Stop when $D \geq R$. ⊡

In step 2, the choice between using a RWM step and a prefetching step is essentially determined by the properties of the available hardware. Prefetching is preferable on theoretical grounds but in practise its performance is more sensitive to the processor-network balance of the parallel computer.

If adaptation is much less frequent than the (prefetching) RWM step, i.e. $K >> M$ which is expected with batch adaptation samplers, there is a better way to implement the algorithm.

Algorithm 2 *Parallel adaptive and hybrid Metropolis-Hastings sampler (the case when $K >> M$)*

1. The current state of the chain is θ_r and the draw counter has the value $D = r$ when the $(r+1)^{th}$ iteration begins and $r = i^A_{j-1}$.

2. (Independence step) Each process p, $p = 1, ..., P$, generates $\theta_p = \left\{\theta_{1p}, \theta_{2p}, ..., \theta_{R_p p}\right\}$ where
$$\theta_{ip} \sim f_j, i = 1, ..., R_p,$$
and collects the values of the posterior evaluated at these parameters in the vector $p_p = \left\{p\left(\theta_{1p}\right), p\left(\theta_{2p}\right), ..., p\left(\theta_{R_p p}\right)\right\}$ (parallel). Here R_p is chosen such that
$$\Sigma^P_{p=1} R_p = i^A_j - i^A_{j-1}.$$
(parallel)

3. The master process gathers θ_p and p_p, $p = 1, ..., P$ (*gather*).

4. (Metropolis-Hastings step) Run the Metropolis-Hastings algorithm with the posterior already evaluated at $\Sigma^P_{p=1} R_p$ parameter values and insert (prefetching) RWM steps as desired (serial).

5. Step 1 to 4 produced $\tilde{D}_{pf} + \Sigma^P_{p=1} R_p$ draws where $\tilde{D}_{pf}$ is the number of draws produced by the (prefetching) RWM steps in 4. Update the draw counter $D = D + \tilde{D}_{pf} + \Sigma^P_{p=1} R_p$.

6. Perform adaptation of the independence proposal density f and let $j = j + 1$ (serial or parallel).

7. Return to 1. Stop when $D \geq R$. ⊡

We note that the algorithms, as stated here, imply that the adaptation points in [14] are stochastic, due to the prefetching RWM step. It is possible, at the expense of more involved notation, to write the algorithms with deterministic adaptation points. However, in practise this does not affect the performance of the algorithm.

Now, let $1 - \alpha_1$ denote the fraction of draws generated by the independence proposal and assume that the parallel computer is homogeneous, such that $R_p = \tilde{R}$ is the same for all processors, and $\Sigma^P_{p=1} R_p = P\tilde{R}$. Under the assumption that the posterior evaluation accounts for the dominant fraction of computational time *and* abstracting from communication and adaptation costs the speedup of the hybrid algorithm is

$$S_h(P, \alpha_1) = \frac{1}{\frac{1-\alpha_1}{P} + \frac{\alpha_1}{S_{pf}(P)}} \tag{15}$$

and

$$1 = S_{rwm}(P) \leq S_{pf}(P) \leq S_h(P, \alpha_1) \leq S_i(P) = P,$$

where $S_{pf}(P) = S_h(P, 1)$ is the speedup of the always reject RWM prefetching algorithm and $S_i(P) = S_h(P, 0) = P$ is the speedup of the pure (adaptive) independence sampler. The speedup of a hybrid algorithm with a RWM step (instead of a prefetching RWM step) is obtained by replacing $S_{pf}(P)$ by $S_{rwm}(P) = 1$ in [15]. In figure 2 in the main text the relative efficiency contours

$$RE_h(P, \alpha_1) = \frac{S_h(P, \alpha_1)}{P} = \frac{1}{1 - \alpha_1 + \frac{P\alpha_1}{S_{pf}(P)}}$$

are displayed. The speedup $S_{pf}(P)$ used in constructing the figure is the optimal speedup of the static prefetching algorithm, under the assumption that the posterior is multivariate normal. Our focus on the always reject variant of prefetching RWM in this paper follows from the idea that the primary role of the RWM steps in the hybrid algorithm is to facilitate the escape from difficult points in the parameter space. For more details on the prefetching RWM approach, see Strid (2009). Note that the optimal acceptance rate of the static prefetching algorithm implies the always reject tour, i.e. it is optimal to allocate processors according to [12].

References

Adolfson, M., Laséen, S., Lindé, J. and Villani, M. (2007), 'Bayesian Estimation of an Open Economy DSGE Model with Incomplete Pass-Through', *Journal of International Economics* **72**, 481–511.

Adolfson, M., Laséen, S., Lindé, J. and Villani, M. (2008*a*), 'Empirical Properties of Closed and Open Economy DSGE Models of the Euro Area', *Macroeconomic Dynamics* **12**, 2–19.

Adolfson, M., Laséen, S., Lindé, J. and Villani, M. (2008*b*), 'Evaluating an Estimated New Keynesian Small Open Economy Model', *Journal of Economic Dynamics and Control* **32**, 2690–2721.

Adolfson, M., Lindé, J. and Villani, M. (2007*a*), 'Bayesian Inference in DSGE Models - Some Comments', *Econometric Reviews* **72**, 173–185.

Adolfson, M., Lindé, J. and Villani, M. (2007*b*), 'Forecasting Performance of an Open Economy DSGE Model', *Econometric Reviews* **26**, 289–328.

An, S. (2005), 'Bayesian Estimation of DSGE Models: Lessons from Second-order Approximations'. Working Paper, University of Pennsylvania.

An, S. and Schorfheide, F. (2007*a*), 'Bayesian Analysis of DSGE Models', *Econometric Reviews* **26(2-4)**, 113–172.

An, S. and Schorfheide, F. (2007*b*), 'Bayesian Analysis of DSGE Models - Rejoinder', *Econometric Reviews* **26(2-4)**, 211–219.

Anderson, G. (2008), 'Solving Linear Rational Expectations Models: A Horse Race', *Computational Economics* **31(2)**, 95–113.

Anderson, G. and Moore, G. (1985), 'A Linear Algebraic Procedure for Solving Linear Perfect Foresight Models', *Economics Letters* **17(3)**, 247–252.

Andreasen, M. M. (2008), 'How to Maximize the Likelihood Function of a DSGE Model', *Computational Economics* . Forthcoming.

Arulampalan, S., Maskell, S., Gordon, N. and Clapp, T. (2002), 'A Tutorial on Particle Filters for On-Line Non-Linear/Non-Gaussian Bayesian Tracking', *IEEE Transactions on Signal Processing* **50**, 174–188.

Brockwell, A. (2006), 'Parallel Markov Chain Monte Carlo Simulation by Pre-fetching', *Journal of Computational and Graphical Statistics* **15**, 246–261.

Brockwell, A. E. and Kadane, J. B. (2005), 'Identification of regeneration times in mcmc simulation, with application to adaptive schemes', *Journal of Computational and Graphical Statistics* **14(2)**, 436–458.

Chib, S. and Ramamurthy, S. (2009), 'Tailored Randomized-block MCMC Methods for Analysis of DSGE Models', *Journal of Econometrics* . Forthcoming.

del Negro, M., Schorfheide, F., Smets, F. and Wouters, R. (2005), 'On the Fit and Forecasting Performance of New-Keynesian Models'. Working Paper Series, No. 491, European Central Bank.

Durbin, J. and Koopman, S. J. (2001), *Time Series Analysis by State Space Methods*, Oxford University Press.

Fernández-Villaverde, J. and Rubio-Ramírez, J. F. (2007), 'Estimating Macroeconomic Models: A Likelihood Approach', *Review of Economic Studies* **74**, 1059–1087.

Haario, H., Saksman, E. and Tamminen, J. (2001), 'An adaptive Metropolis algorithm', *Bernoulli* **7(2)**, 223–242.

Harvey, A. (1989), *Forecasting, Structural Time Series Models and the Kalman filter*, Cambridge University Press.

Klein, P. (2000), 'Using the Generalized Schur Form to Solve a Multivariate Linear Rational Expectations Model', *Journal of Economic Dynamics and Control* **24**, 1405–1423.

Kohn, R. and Giordani, P. (2008), 'Adaptive Independent Metropolis-Hastings by Fast Estimation of Mixtures of Normals', *Journal of Computational Graphics and Statistics* . Forthcoming.

Lubik, T. A. and Schorfheide, F. (2004), 'Testing for Indeterminacy: An Application to U.S. Monetary Policy', *American Economic Review* **94**(1), 190–217.
URL: *http://ideas.repec.org/a/aea/aecrev/v94y2004i1p190-217.html*

Roberts, G. O. and Rosenthal, J. S. (2001), 'Optimal Scaling for Various Random Walk Metropolis Algorithms', *Statistical Science* **16(4)**, 351–367.

Schmitt-Grohe, S. and Uribe, M. (2004), 'Solving Dynamic General Equilibrium Models Using a Second-Order Approximation to the Policy Function', *Journal of Economic Dynamics and Control* **28**, 755–775.

Silva, R., Kohn, R., Giordani, P. and Mun, X. (2008), 'A Copula Based Approach to Adaptive Sampling'.

Sims, C. A. (1999), Matlab Optimization Software. Code package.

Sims, C. A. (2000), Solving linear rational expectations models. Mimeo, Princeton University.

Smets, F. and Wouters, R. (2003), 'An Estimated Stochastic Dynamic General Equilibrium Model of the Euro Area', *Journal of the European Economic Association* **20**, 891–910.

Strid, I. (2009), 'Efficient Parallelisation of Metropolis-Hastings Algorithms Using a Prefetching Approach', *Computational Statistics and Data Analysis* . Forthcoming.

Strid, I. and Walentin, K. (2008), 'Block Kalman Filters for Large-scale DSGE models', *Computational Economics* **33(3)**, 277–304.

Whiley, M. and Wilson, S. P. (2004), 'Parallel Algorithms for Markov Chain Monte Carlo in Latent Spatial Gaussian Models', *Statistics and Computing* **14**, 171–179.

Wilkinson, D. (2006), Parallel Bayesian Computation, *in* E. J. Kontoghiorghes, ed., 'Handbook of Parallel Computing and Statistics', Chapman and Hall, chapter 16, pp. 477–508.